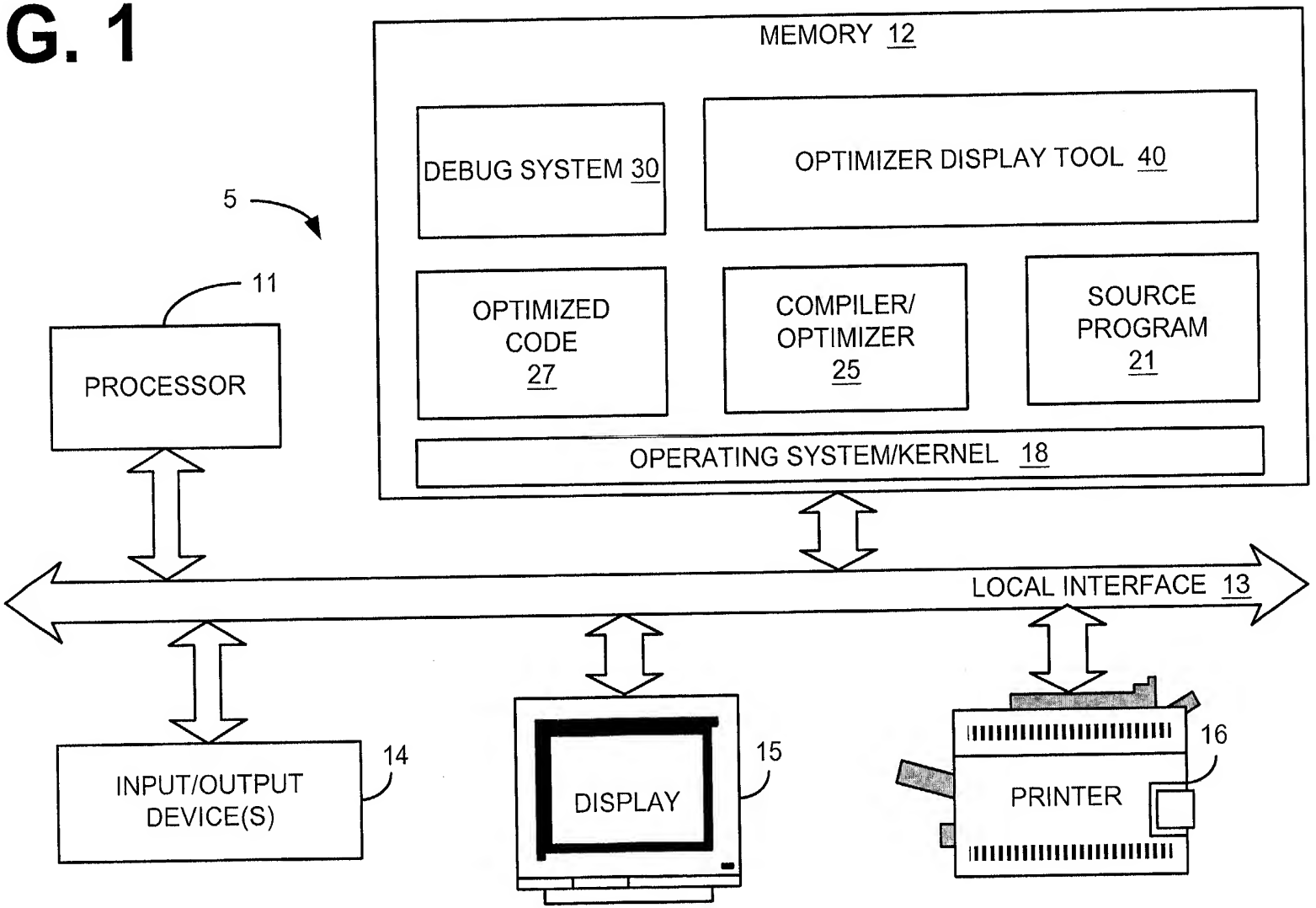
20060720.2003

FIG. 1



20T0E0*Z0HZ800T

FIG. 2

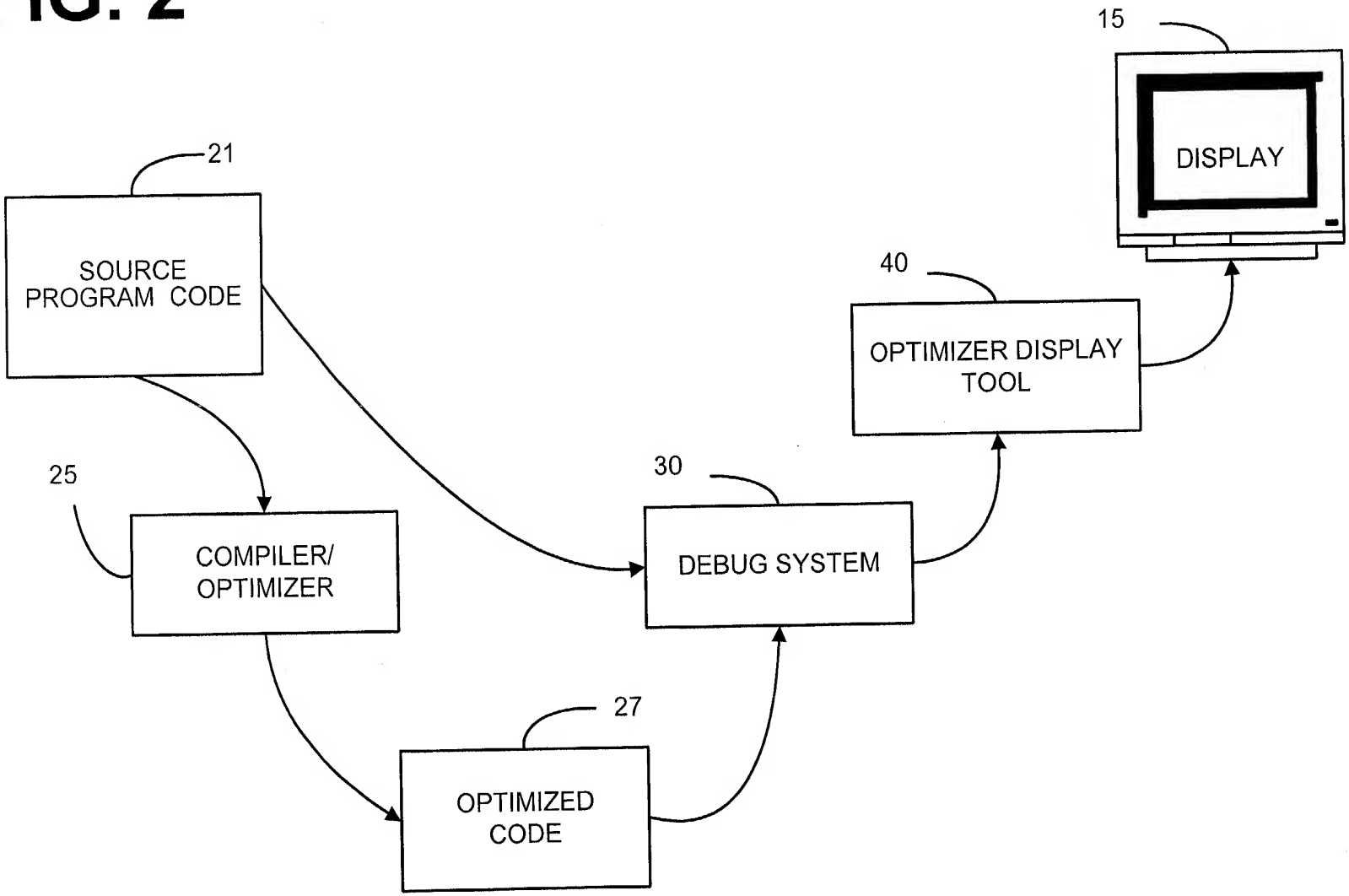
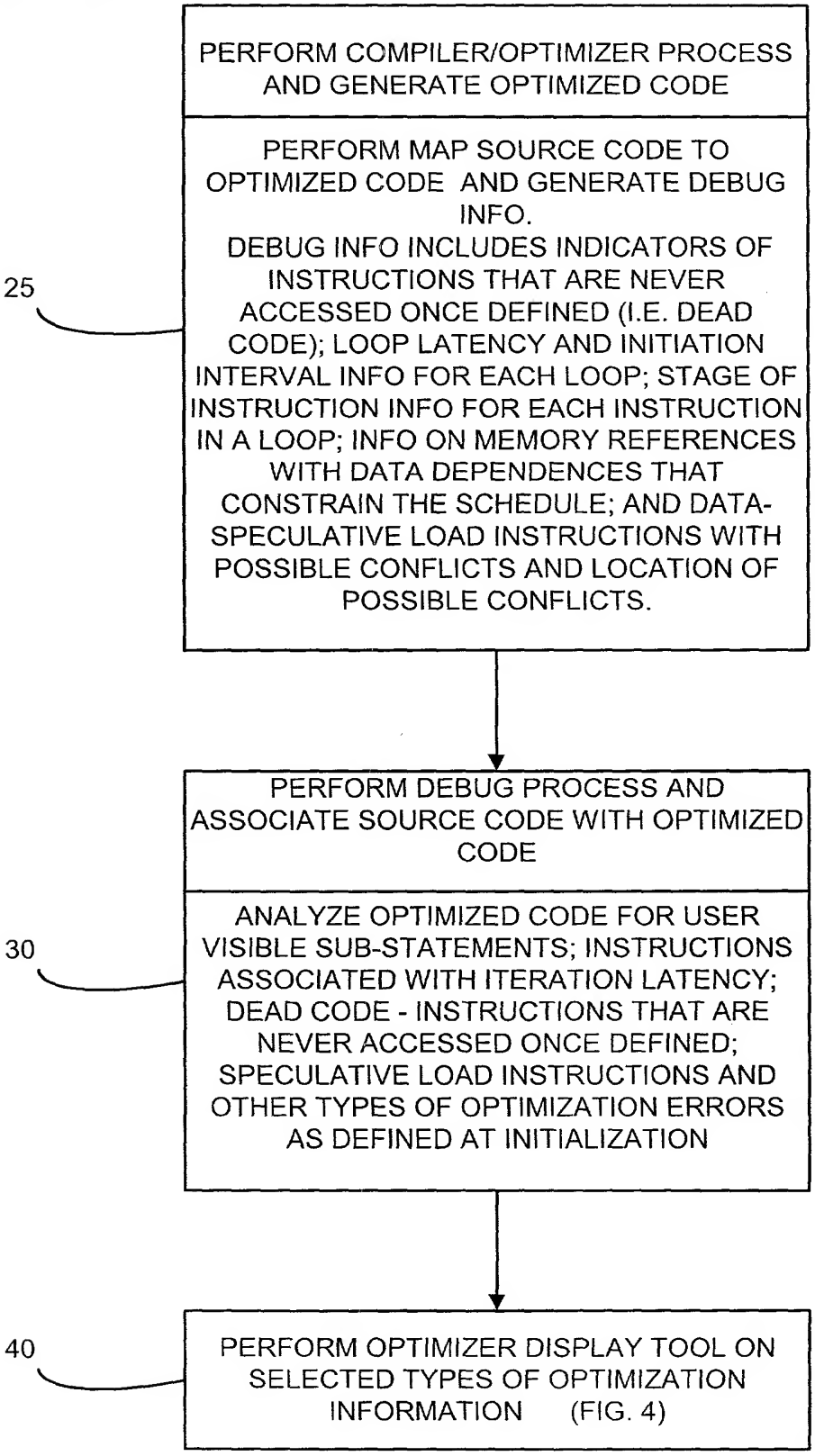


FIG. 3



20706.0054.00

FIG. 4

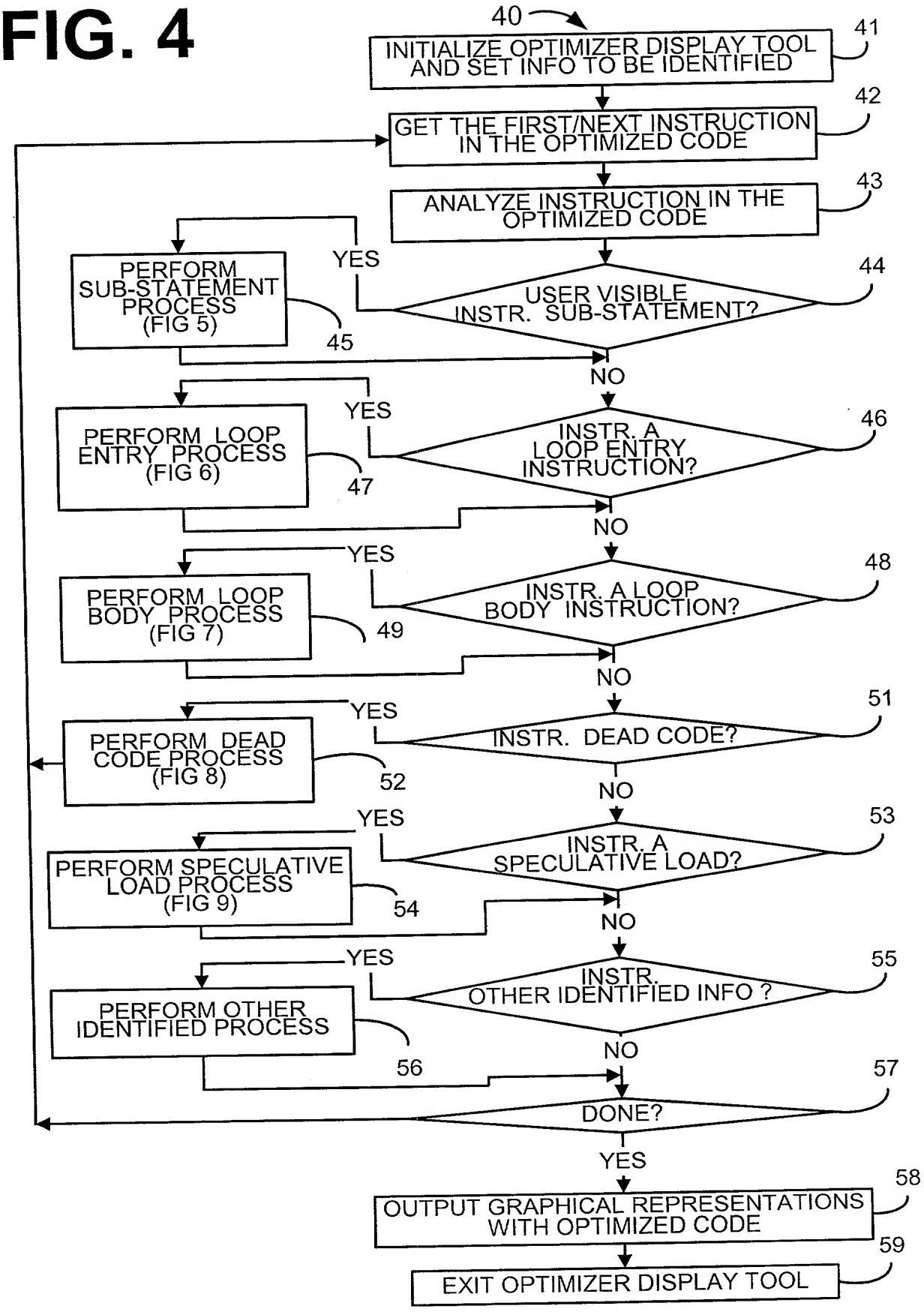

40
INITIALIZE OPTIMIZER DISPLAY TOOL AND SET INFO TO BE IDENTIFIED
41
GET THE FIRST/NEXT INSTRUCTION IN THE OPTIMIZED CODE
42
ANALYZE INSTRUCTION IN THE OPTIMIZED CODE
43
USER VISIBLE INSTR. SUB-STATEMENT?
44
YES
PERFORM SUB-STATEMENT PROCESS (FIG 5)
45
NO
INSTR. A LOOP ENTRY INSTRUCTION?
46
YES
PERFORM LOOP ENTRY PROCESS (FIG 6)
47
NO
INSTR. A LOOP BODY INSTRUCTION?
48
YES
PERFORM LOOP BODY PROCESS (FIG 7)
49
NO
INSTR. DEAD CODE?
51
YES
PERFORM DEAD CODE PROCESS (FIG 8)
52
NO
INSTR. A SPECULATIVE LOAD?
53
YES
PERFORM SPECULATIVE LOAD PROCESS (FIG 9)
54
NO
INSTR. OTHER IDENTIFIED INFO ?
55
YES
PERFORM OTHER IDENTIFIED PROCESS
56
NO
DONE?
57
YES
OUTPUT GRAPHICAL REPRESENTATIONS WITH OPTIMIZED CODE
58
EXIT OPTIMIZER DISPLAY TOOL
59

FIG. 5

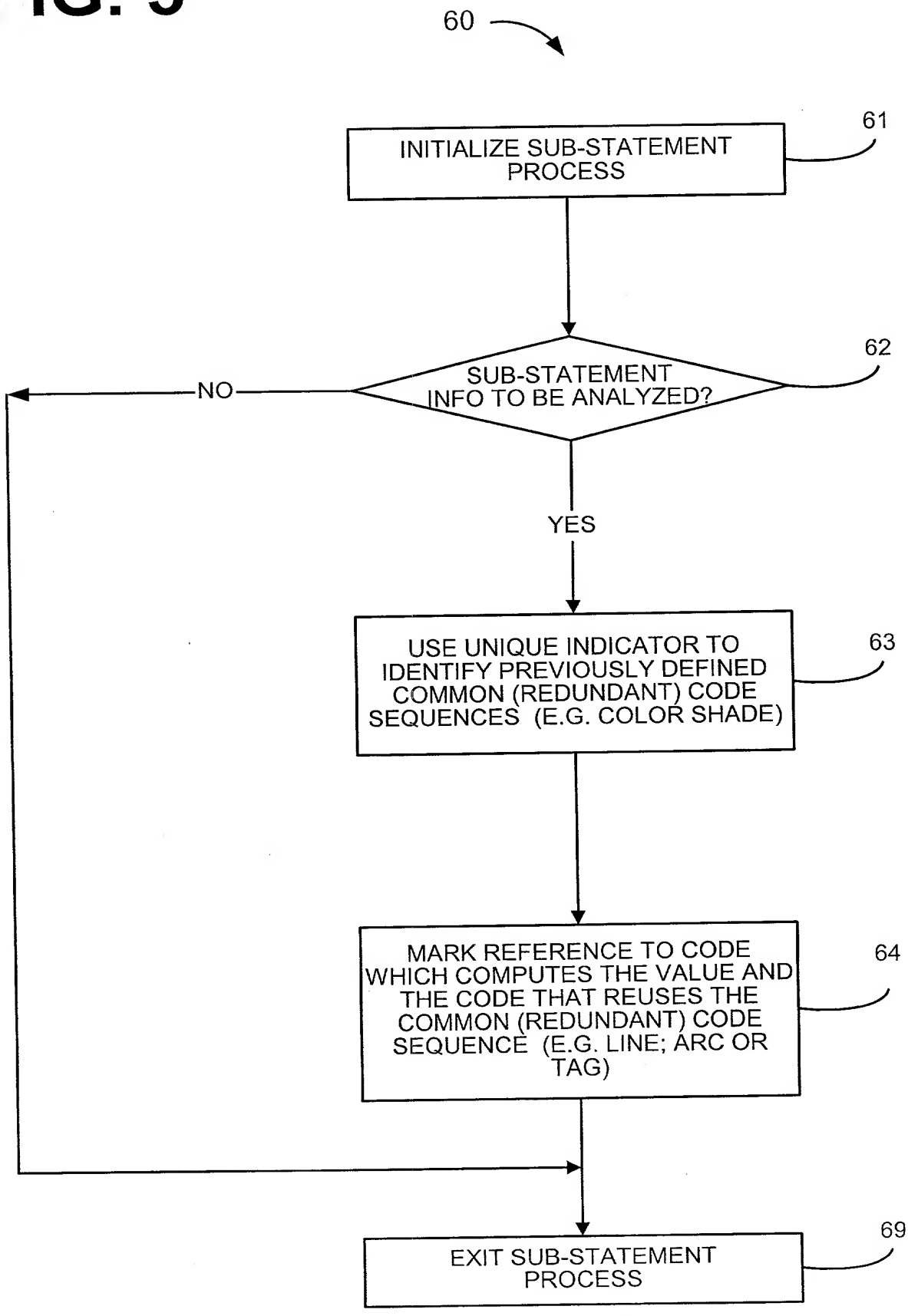
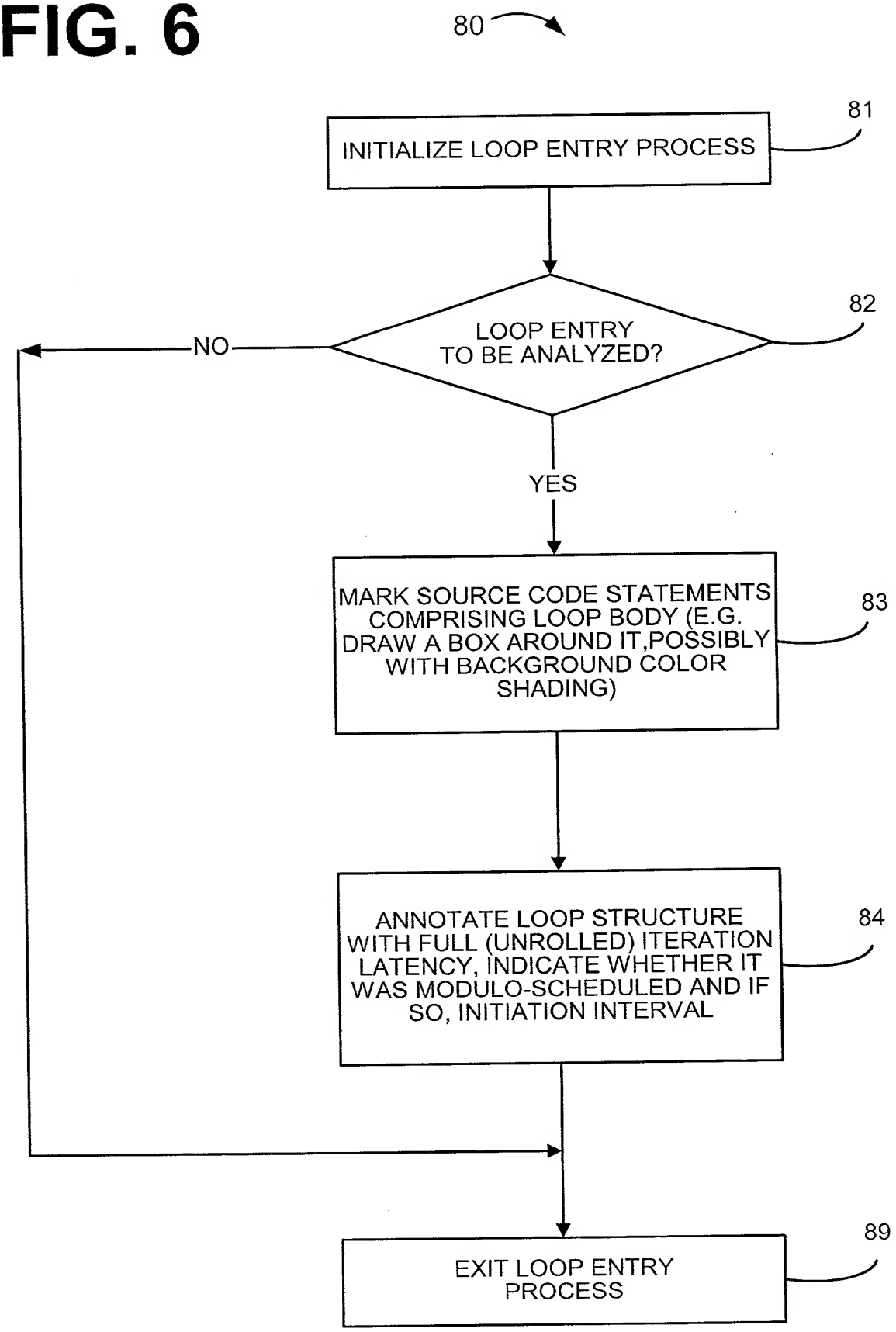

60
INITIALIZE SUB-STATEMENT PROCESS
61
SUB-STATEMENT INFO TO BE ANALYZED?
62
NO
YES
USE UNIQUE INDICATOR TO IDENTIFY PREVIOUSLY DEFINED COMMON (REDUNDANT) CODE SEQUENCES (E.G. COLOR SHADE)
63
MARK REFERENCE TO CODE WHICH COMPUTES THE VALUE AND THE CODE THAT REUSES THE COMMON (REDUNDANT) CODE SEQUENCE (E.G. LINE; ARC OR TAG)
64
EXIT SUB-STATEMENT PROCESS
69

FIG. 6



SOFTSEC 2004-0001

FIG. 7

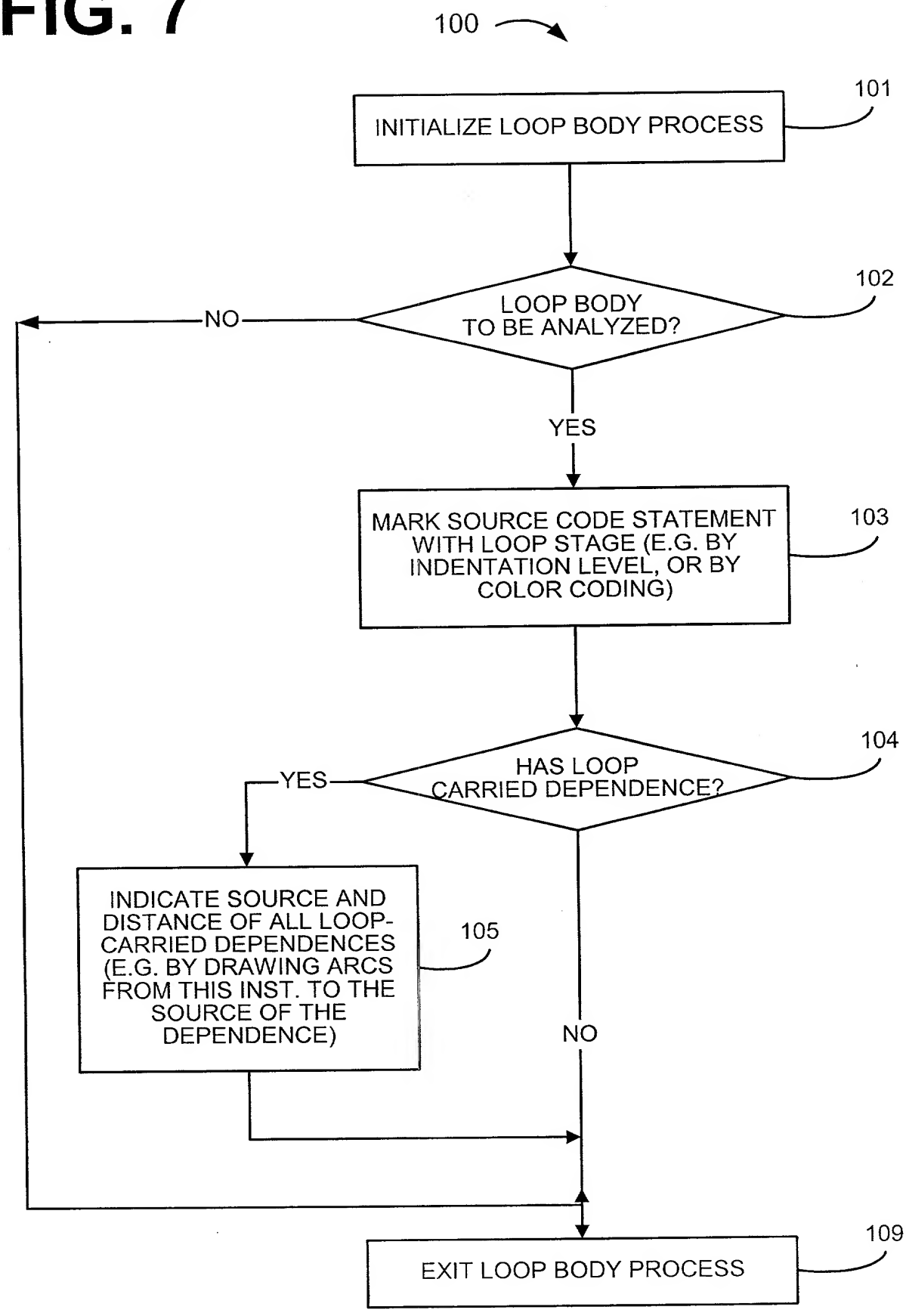

100
INITIALIZE LOOP BODY PROCESS
101
LOOP BODY TO BE ANALYZED?
102
NO
YES
MARK SOURCE CODE STATEMENT WITH LOOP STAGE (E.G. BY INDENTATION LEVEL, OR BY COLOR CODING)
103
HAS LOOP CARRIED DEPENDENCE?
104
YES
NO
INDICATE SOURCE AND DISTANCE OF ALL LOOP-CARRIED DEPENDENCES (E.G. BY DRAWING ARCS FROM THIS INST. TO THE SOURCE OF THE DEPENDENCE)
105
EXIT LOOP BODY PROCESS
109

FIG. 8

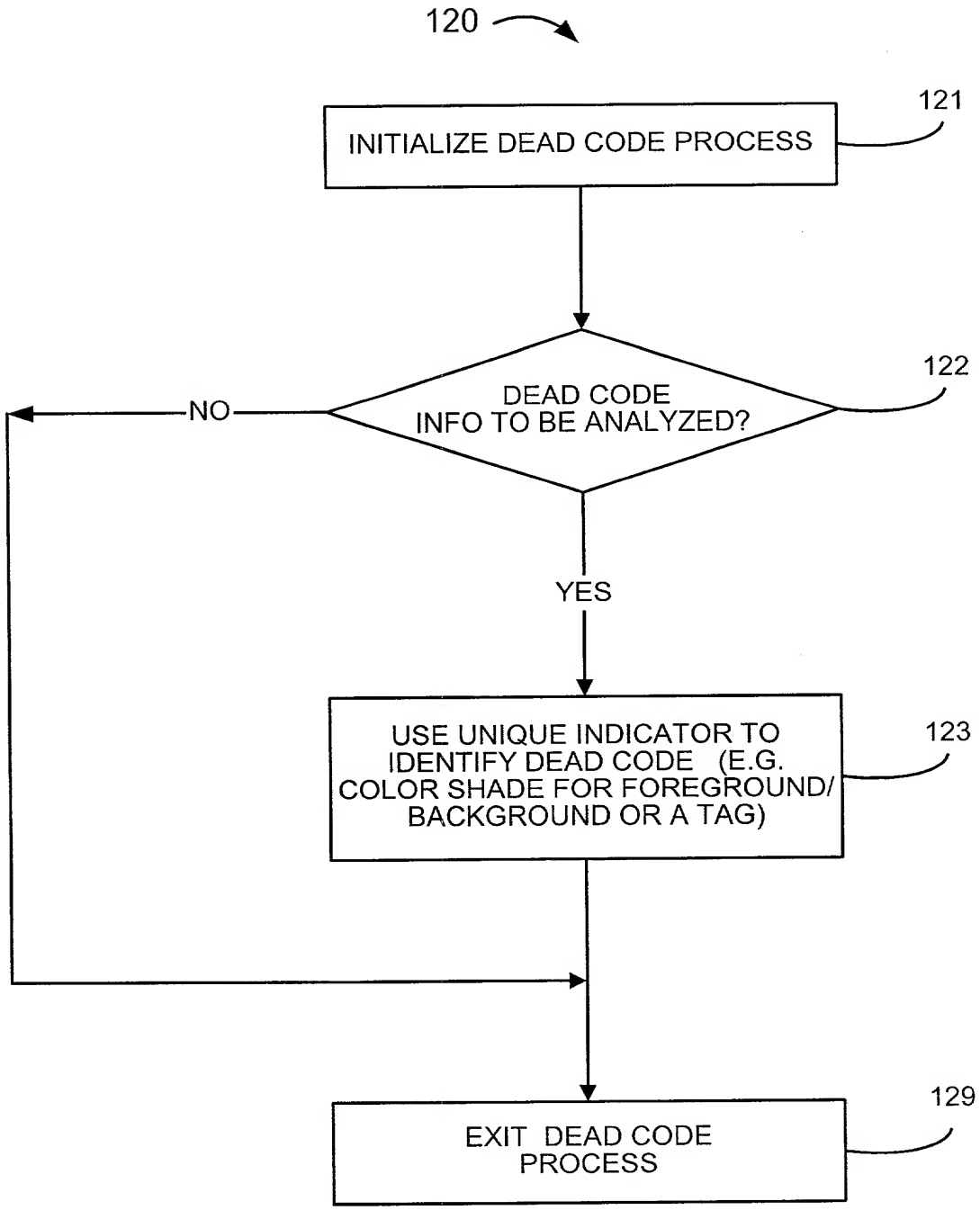

120
INITIALIZE DEAD CODE PROCESS
121
DEAD CODE INFO TO BE ANALYZED?
122
NO
YES
USE UNIQUE INDICATOR TO IDENTIFY DEAD CODE (E.G. COLOR SHADE FOR FOREGROUND/ BACKGROUND OR A TAG)
123
EXIT DEAD CODE PROCESS
129

FIG. 9

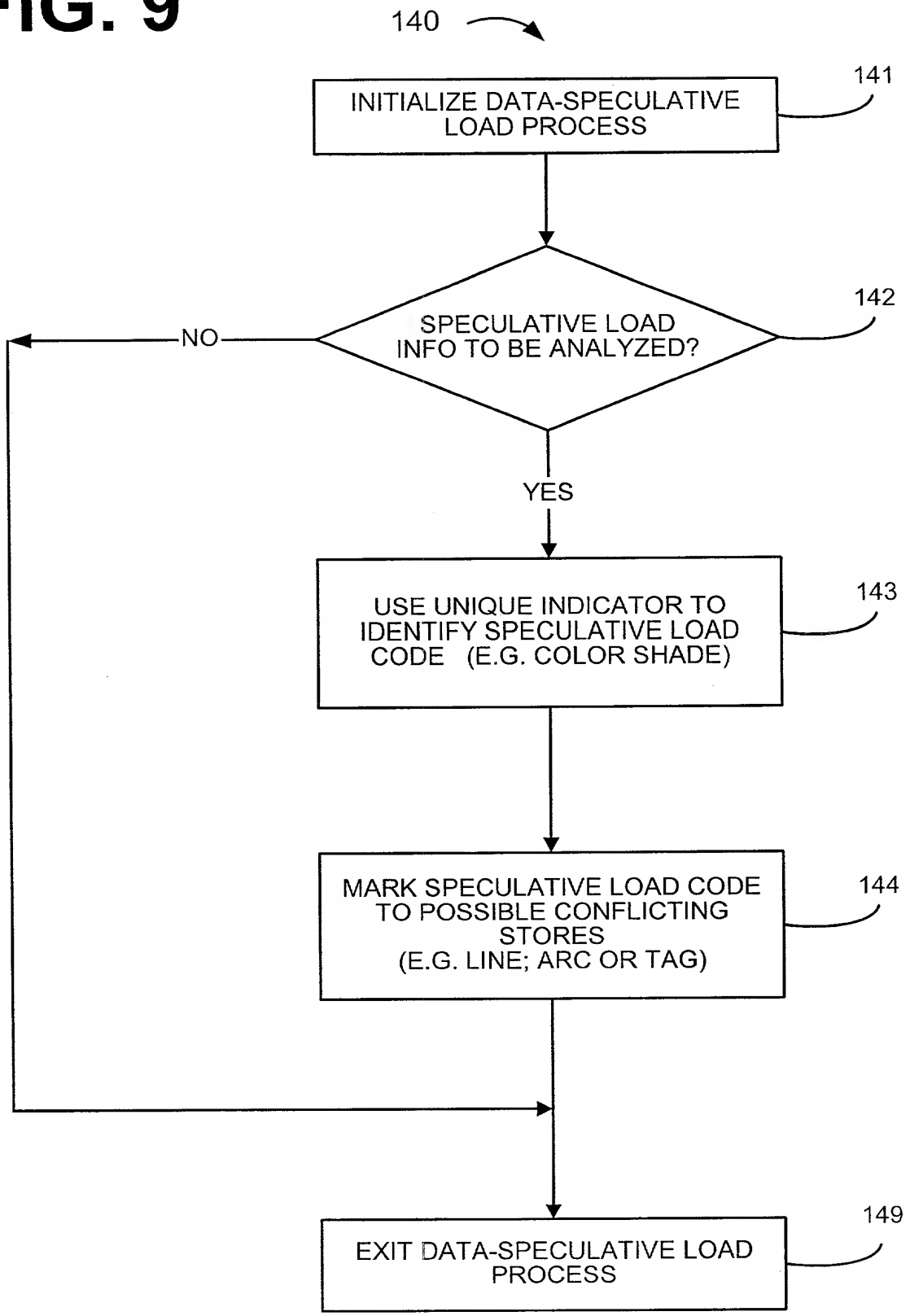
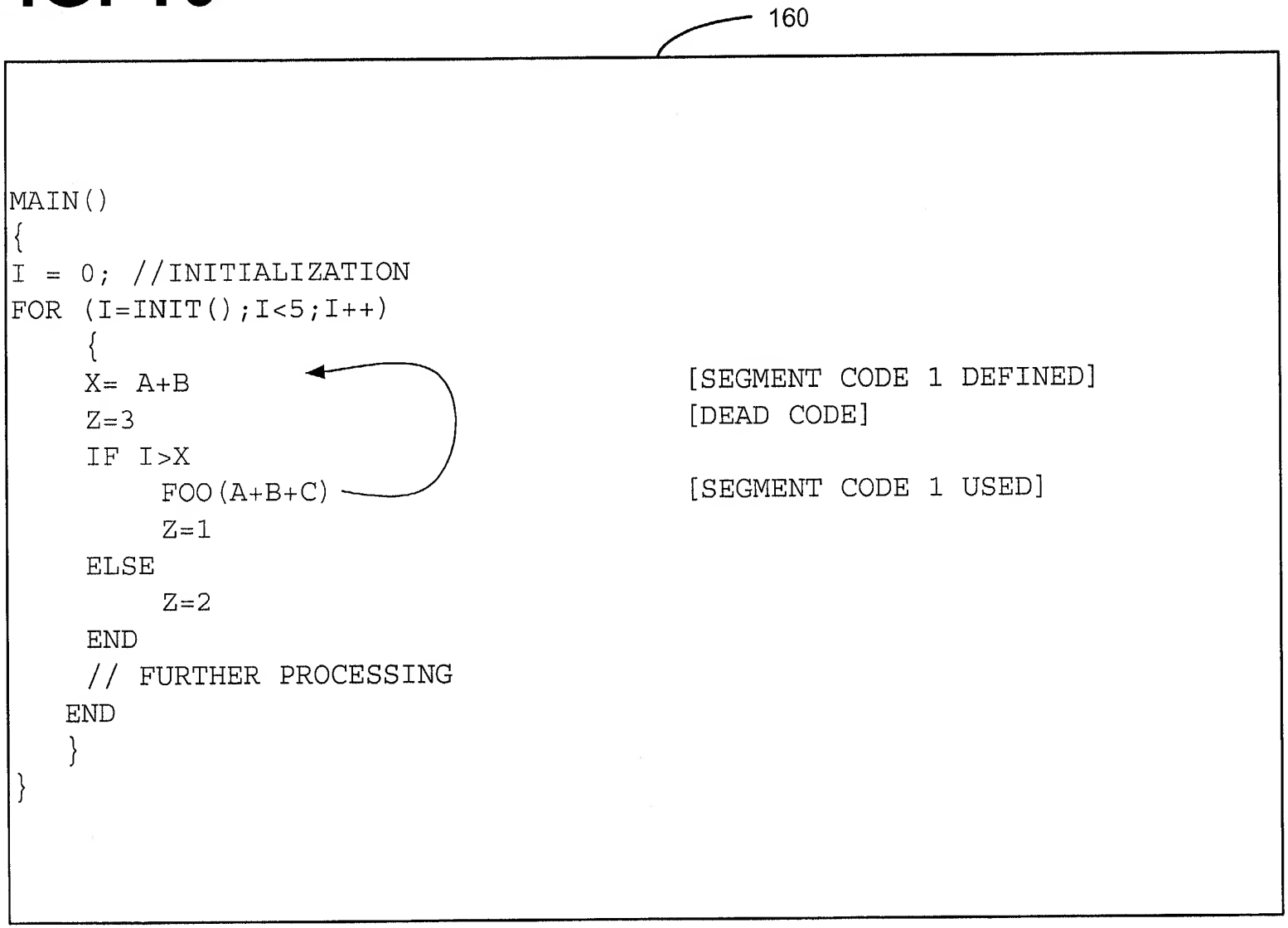

140
INITIALIZE DATA-SPECULATIVE LOAD PROCESS
141
SPECULATIVE LOAD INFO TO BE ANALYZED?
142
NO
YES
USE UNIQUE INDICATOR TO IDENTIFY SPECULATIVE LOAD CODE (E.G. COLOR SHADE)
143
MARK SPECULATIVE LOAD CODE TO POSSIBLE CONFLICTING STORES (E.G. LINE; ARC OR TAG)
144
EXIT DATA-SPECULATIVE LOAD PROCESS
149

FIG. 10


160
MAIN()
{
I = 0; //INITIALIZATION
FOR (I=INIT();I<5;I++)
{
X= A+B
Z=3
IF I>X
FOO(A+B+C)
Z=1
ELSE
Z=2
END
// FURTHER PROCESSING
END
}
}
[SEGMENT CODE 1 DEFINED]
[DEAD CODE]
[SEGMENT CODE 1 USED]